<https://www.youtube.com/watch?v=ReauId6jFFI>

Generally you iterate through the animation enumerable to get at the AnimationState objects:

1. foreach(AnimationState state in animation)
2. {
3. //do something with state
4. }

You could also add them all into a list fairly easily, using linq to cast the enumerable (or use an arraylist instead i guess)

1. List<AnimationState> states = new List<AnimationState>(animation.Cast<AnimationState>());

or

1. ArrayList states = new ArrayList(animation);

From the state, you can get at the name for playing, or even grab the clip it refers to (.name and .clip respectively)

[Add comment](https://answers.unity.com/questions/63138/how-to-get-a-reference-of-animations-array.html) ·  Hide 1 · [Share](https://answers.unity.com/questions/63138/how-to-get-a-reference-of-animations-array.html)

[avatar imagechanfort](https://answers.unity.com/users/271298/chanfort.html)· [Mar 01, 2016 at 10:04 PM](https://answers.unity.com/comments/1149332/view.html) 0

5 years old trick and still works... If you use just

foreach(AnimationState state in animation)

{

//do something with state

}

there are various surprises coming, like "cannot cast from source to destination" or the last state name is somehow the same as the first one. The correct solution seems to be:

List<AnimationState> states = new List<AnimationState>(animation.Cast<AnimationState>());

foreach(AnimationState state in states)

{

//do something with state

}

[avatar image](https://answers.unity.com/users/475709/mightyjoe1.html)

**5**

**Answer** by [MightyJoe1](https://answers.unity.com/users/475709/mightyjoe1.html) · Jan 12, 2015 at 10:19 PM

This code should work with C# to print all the AnimationClip names.

Just pass in a gameObject with an Animation Component.

1. public void PrintAnimations(GameObject character)
2. {
3. Animation anim = character.GetComponent<Animation>();
4. foreach(AnimationState state in anim)
5. {
6. Debug.Log(state.name);
7. }
8. }

Also you could play the animation by calling ...

...

1. anim.Play(state.name);